



US008224784B2

(12) **United States Patent**
Henrickson

(10) **Patent No.:** **US 8,224,784 B2**
(45) **Date of Patent:** **Jul. 17, 2012**

(54) **COMBINED COMPUTER DISASTER RECOVERY AND MIGRATION TOOL FOR EFFECTIVE DISASTER RECOVERY AS WELL AS THE BACKUP AND MIGRATION OF USER- AND SYSTEM-SPECIFIC INFORMATION**

6,202,073 B1 3/2001 Takahashi
6,219,669 B1 4/2001 Haff et al.
6,324,690 B1 11/2001 Luu
6,449,733 B1 9/2002 Bartlett et al.
6,535,915 B1 3/2003 Valys et al.
6,598,223 B1 7/2003 Vrhel, Jr. et al.
6,611,923 B1 8/2003 Mutalik et al.

(Continued)

(75) Inventor: **David L. Henrickson**, Hialeah, FL (US)

FOREIGN PATENT DOCUMENTS

(73) Assignee: **Microsoft Corporation**, Redmond, WA (US)

WO 0153938 A1 7/2001

OTHER PUBLICATIONS

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 600 days.

Microsoft Technet, "Migrating Files and Settings: A Step-by-Step Guide", Dated Jun. 1, 2001, pp. 1-8.

(Continued)

(21) Appl. No.: **10/917,637**

(22) Filed: **Aug. 13, 2004**

Primary Examiner — John E Breene

Assistant Examiner — Christopher J Raab

(74) *Attorney, Agent, or Firm* — Microsoft Corporation

(65) **Prior Publication Data**

US 2006/0036658 A1 Feb. 16, 2006

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **707/654**; 707/659; 714/4.11; 714/4.12;
711/162; 709/221

(58) **Field of Classification Search** 709/221;
714/4.11, 4.12; 707/654, 659
See application file for complete search history.

(57) **ABSTRACT**

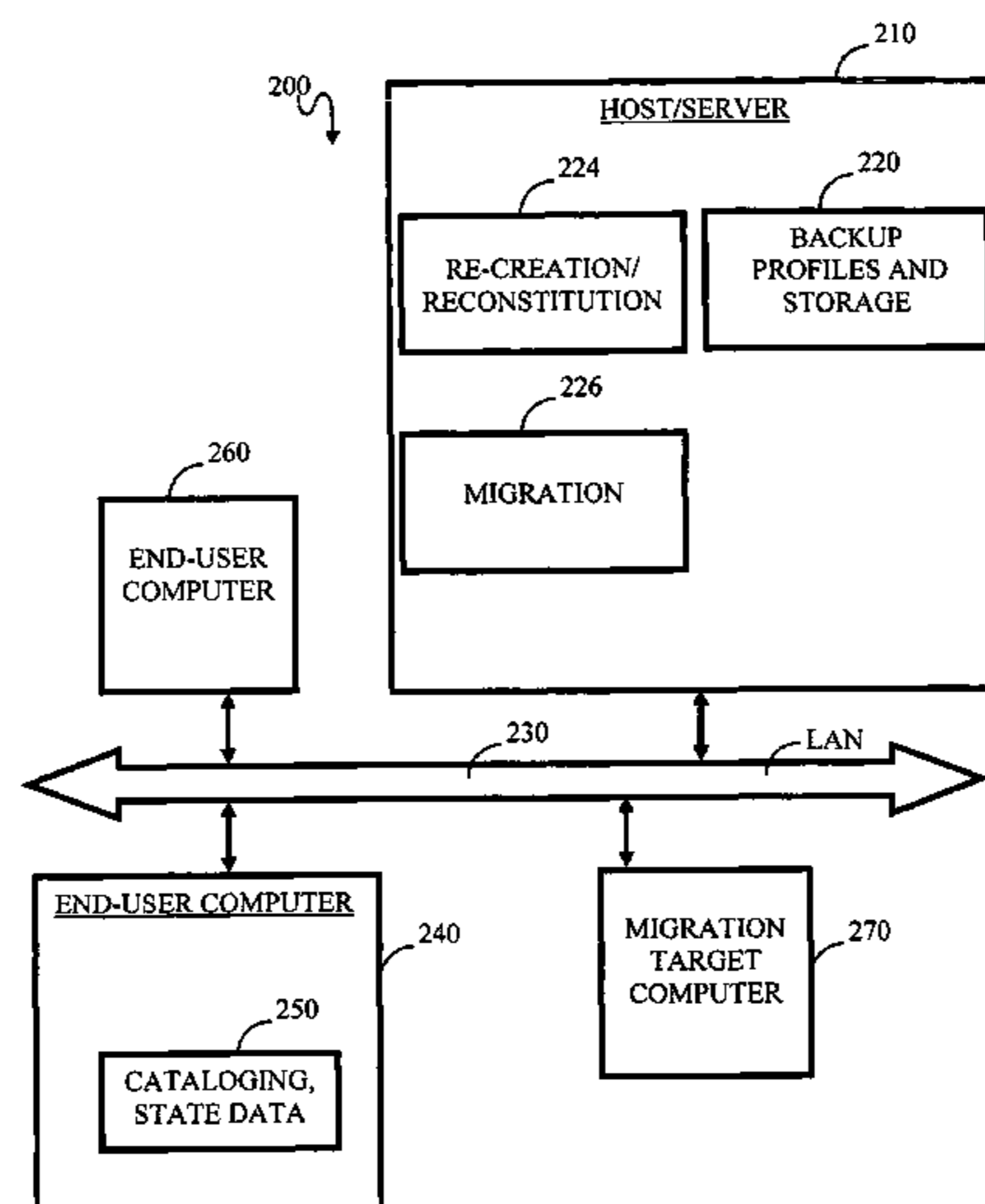
Computer tools and methods novelly combine periodic backup and restore features with migration features to transfer the components of a failed system to a new system. As well as backing up and transferring critical data files during the disaster recovery operation, an aspect of the present invention also transfers, inter alia, applications, user states, hardware settings, software settings, user preferences and other user settings, menus, and directories. In another aspect of the present invention, a network of shared end-user computers periodically backs up each individual end-user computer to a central instrumentality, in a novel manner to reduce storage and time requirements. That is, redundancies for the end-user computer backup files are eliminated or greatly reduced by only making reference to but not storing components that are generic to the system, and further for each previously stored component, only storing differential or incremental information, unless predefined conditions are met.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,958,270 A 9/1990 McLaughlin et al.
5,555,416 A 9/1996 Owens et al.
5,758,150 A 5/1998 Bell et al.
5,819,020 A * 10/1998 Beeler, Jr. 714/5
5,870,611 A 2/1999 London Shrader et al.
5,915,252 A 6/1999 Misheski et al.
6,029,232 A * 2/2000 Honda 711/162
6,192,518 B1 2/2001 Neal

20 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

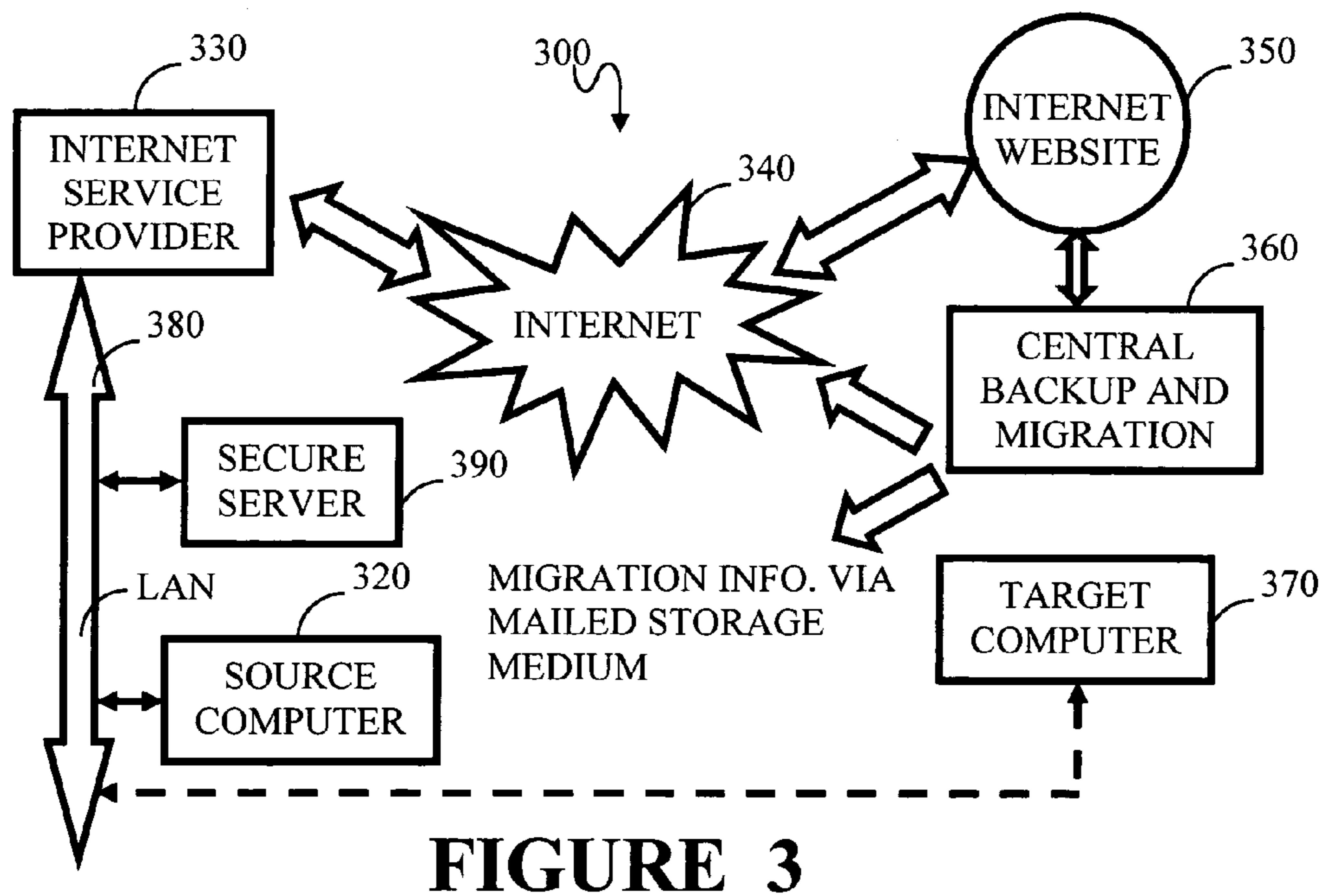
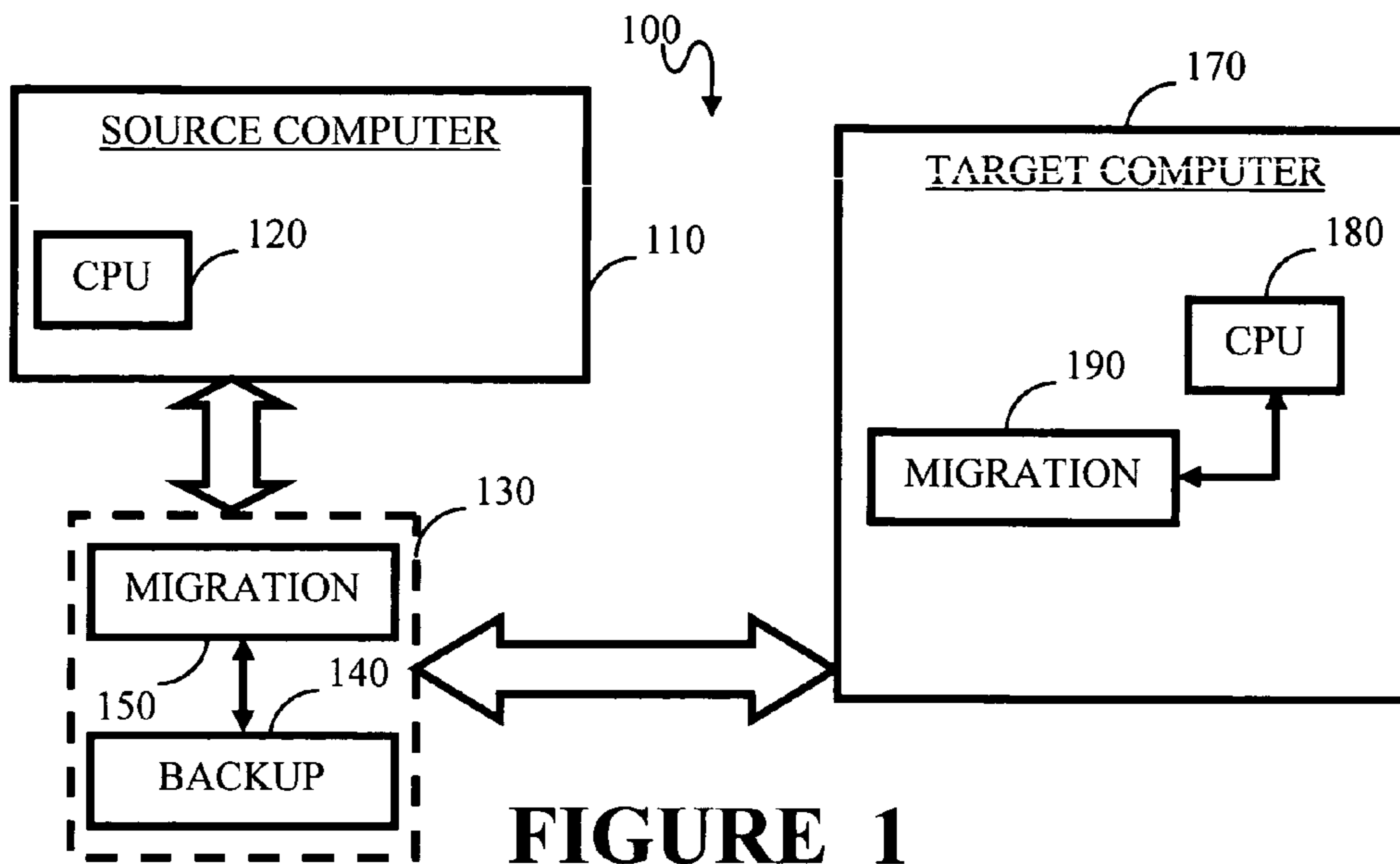
6,625,622 B1 9/2003 Henrickson et al.
 6,671,705 B1* 12/2003 Duprey et al. 707/204
 6,728,877 B2 4/2004 Mackin et al.
 6,760,708 B1 7/2004 Hubbard et al.
 6,853,978 B2 2/2005 Forth et al.
 6,865,655 B1* 3/2005 Andersen 711/162
 6,912,384 B2 6/2005 Huomo et al.
 6,928,644 B1 8/2005 Kroening et al.
 6,931,558 B1* 8/2005 Jeffe et al. 713/340
 6,938,058 B2 8/2005 Henrickson et al.
 6,944,790 B2 9/2005 Hamilton et al.
 6,950,864 B1* 9/2005 Tsuchiya 709/223
 6,963,908 B1 11/2005 Lynch et al.
 7,024,528 B2* 4/2006 LeCrone et al. 711/162
 7,051,327 B1* 5/2006 Milius et al. 717/177
 7,062,645 B2 6/2006 Kroening
 7,069,402 B2 6/2006 Coulter et al.
 7,082,553 B1 7/2006 Wang
 7,085,904 B2* 8/2006 Mizuno et al. 711/162
 7,100,007 B2* 8/2006 Saika 711/162
 7,103,740 B1* 9/2006 Colgrove et al. 711/162
 7,143,096 B2* 11/2006 Gemba et al. 707/10
 7,210,131 B2 4/2007 Schmidt et al.
 7,225,211 B1* 5/2007 Colgrove et al. 707/205
 7,293,133 B1* 11/2007 Colgrove et al. 711/111
 7,313,722 B2* 12/2007 Saika 714/10
 7,330,997 B1* 2/2008 Odom 714/6.23
 2001/0034728 A1* 10/2001 McBride et al. 707/1
 2001/0056438 A1* 12/2001 Ito 707/204
 2002/0010808 A1 1/2002 Wiggins et al.
 2002/0083366 A1 6/2002 Ohran
 2002/0087588 A1* 7/2002 McBride et al. 707/204
 2002/0103779 A1 8/2002 Ricart et al.
 2002/0104080 A1 8/2002 Woodard et al.
 2002/0147938 A1* 10/2002 Hamilton, II et al. 714/6
 2002/0160760 A1 10/2002 Aoyama
 2003/0009754 A1 1/2003 Rowley et al.
 2003/0037327 A1 2/2003 Ciccirelli et al.
 2003/0037328 A1 2/2003 Ciccirelli et al.
 2003/0074386 A1 4/2003 Schmidt et al.
 2003/0105912 A1 6/2003 Noren
 2003/0134625 A1 7/2003 Choi
 2003/0172373 A1 9/2003 Henrickson et al.
 2003/0200542 A1 10/2003 Shaughnessy et al.
 2003/0220991 A1* 11/2003 Soejima et al. 709/221
 2004/0025155 A1 2/2004 Sedlack et al.
 2004/0103064 A1 5/2004 Howard et al.
 2004/0128203 A1 7/2004 Pierre et al.
 2004/0139128 A1* 7/2004 Becker et al. 707/204
 2004/0142684 A1 7/2004 Ratert et al.
 2004/0158766 A1* 8/2004 Liccione et al. 714/4
 2004/0163008 A1* 8/2004 Kim 714/4
 2004/0199812 A1 10/2004 Earl et al.

2004/0235523 A1 11/2004 Schrire et al.
 2004/0237079 A1 11/2004 Cox et al.
 2004/0255179 A1* 12/2004 Mayer 714/1
 2005/0044232 A1 2/2005 Keane
 2005/0125464 A1* 6/2005 Kline 707/204
 2005/0160305 A1* 7/2005 Soejima 714/2
 2005/0191998 A1 9/2005 Onyon et al.
 2005/0246575 A1 11/2005 Chen et al.
 2006/0010436 A1 1/2006 Henrickson et al.
 2006/0036658 A1 2/2006 Henrickson
 2006/0036890 A1 2/2006 Henrickson
 2006/0036895 A1 2/2006 Henrickson
 2006/0183469 A1 8/2006 Gadson
 2007/0067766 A1 3/2007 Tal et al.

OTHER PUBLICATIONS

Microsoft Technet, "User State Migration in Windows XP", Dated Jun. 1, 2001, pp. 1-8.
 Internet website titled "Fusion-one-Mobile & Data Synchronization Solutions" with the world Wide Web address of: <http://www.fusionone.com>.
 Anon., "M-Commerce Initiative Launched by InstantService.com and MobileUSA.com; Live interaction for E-commerce Sales and Service Anywhere," Business Wire, Jul. 19, 2000.
 Deignam, M.P., "Transfer Personal Settings Between Computers," WinMag.com, Nov. 29, 2000.
 Anon., "PictureTaker Enterprise Edition Version 3.0 Enabled for Windows 2000," Business Wire, Feb. 17, 2000.
 Anon., "Tranxition Corporation Announces Availability of Personality Transport Professional," Business Wire, Apr. 3, 2000.
 O'Brien, J., "Aveo Knows What Your PC Is Doing; Finds Software Conflicts," Financial Post, p. 8, Jul. 29, 2000.
 Raskin, R. "Easy Connections from Old to New Software Helps Transfer Data, Programs from PC to PC," Arizona Republic, final chaser edition, business section, p. D3, Sep. 25, 2001.
 Fried, J. J., "Program not as Simple as it Sounds: Alohabob is Supposed to Easily Transfer Data Between Computers, but the Process Isn't Always Quick or Easy," Philadelphia Inquirer, City-D edition, Tech.Life section, p. C07, Jul. 18, 2002.
 Anon., "Miramar Systems' Desktop DNA Utility Chosen by MCI WorldCom for Hardware Refresh Program," Business Wire, p. 0090, Oct. 12, 2000.
 Non-Final Office Action cited in related U.S. Appl. No. 10/917,631 dated Jan. 11, 2007.
 Final Office Action cited in related U.S. Appl. No. 10/917,631 dated Aug. 5, 2008.
 Non-Final Office Action cited in related U.S. Appl. No. 10/917,631 dated Mar. 19, 2009.
 Notice of Allowance cited in related U.S. Appl. No. 10/917,631 dated Sep. 4, 2009.

* cited by examiner



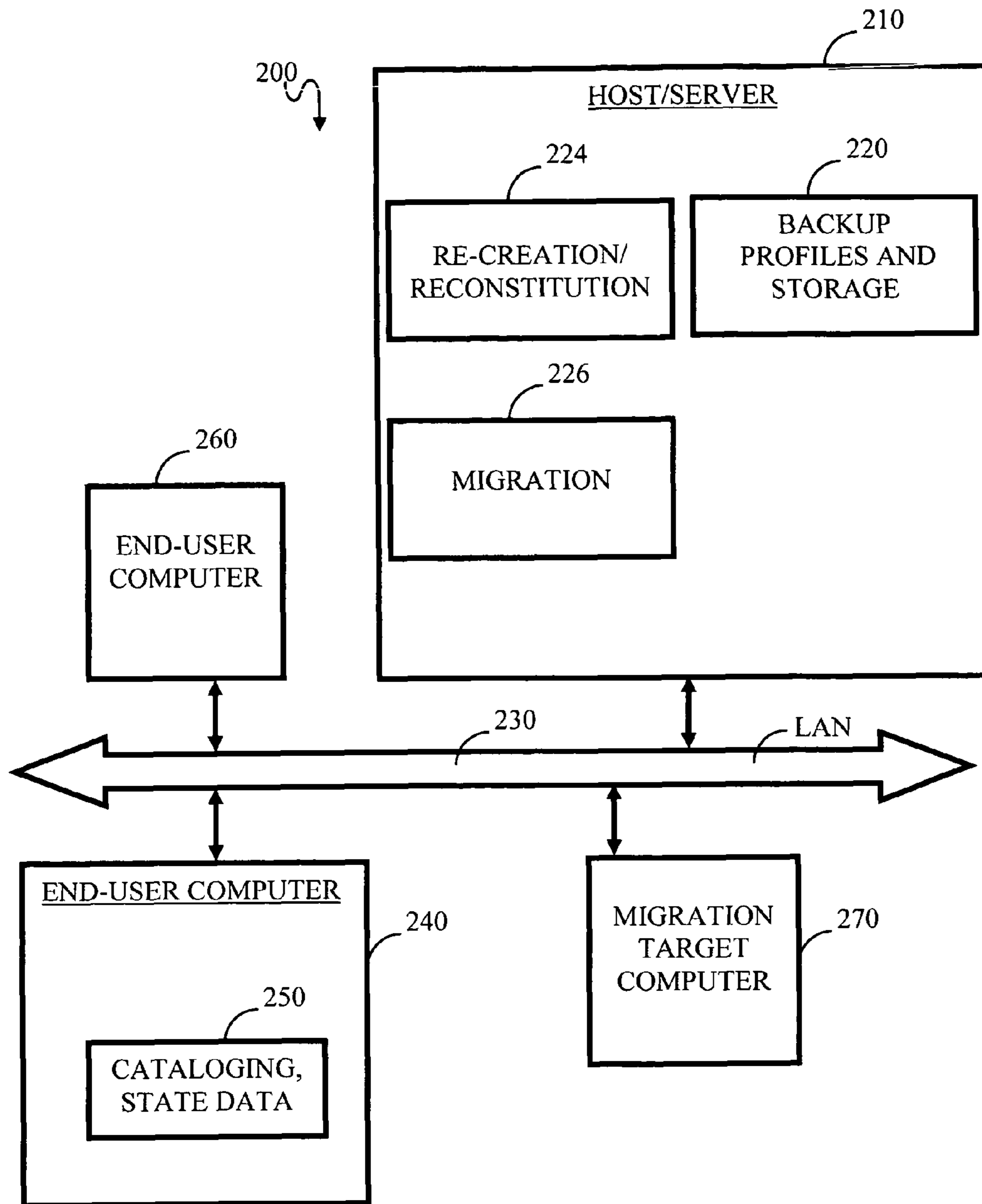


FIGURE 2

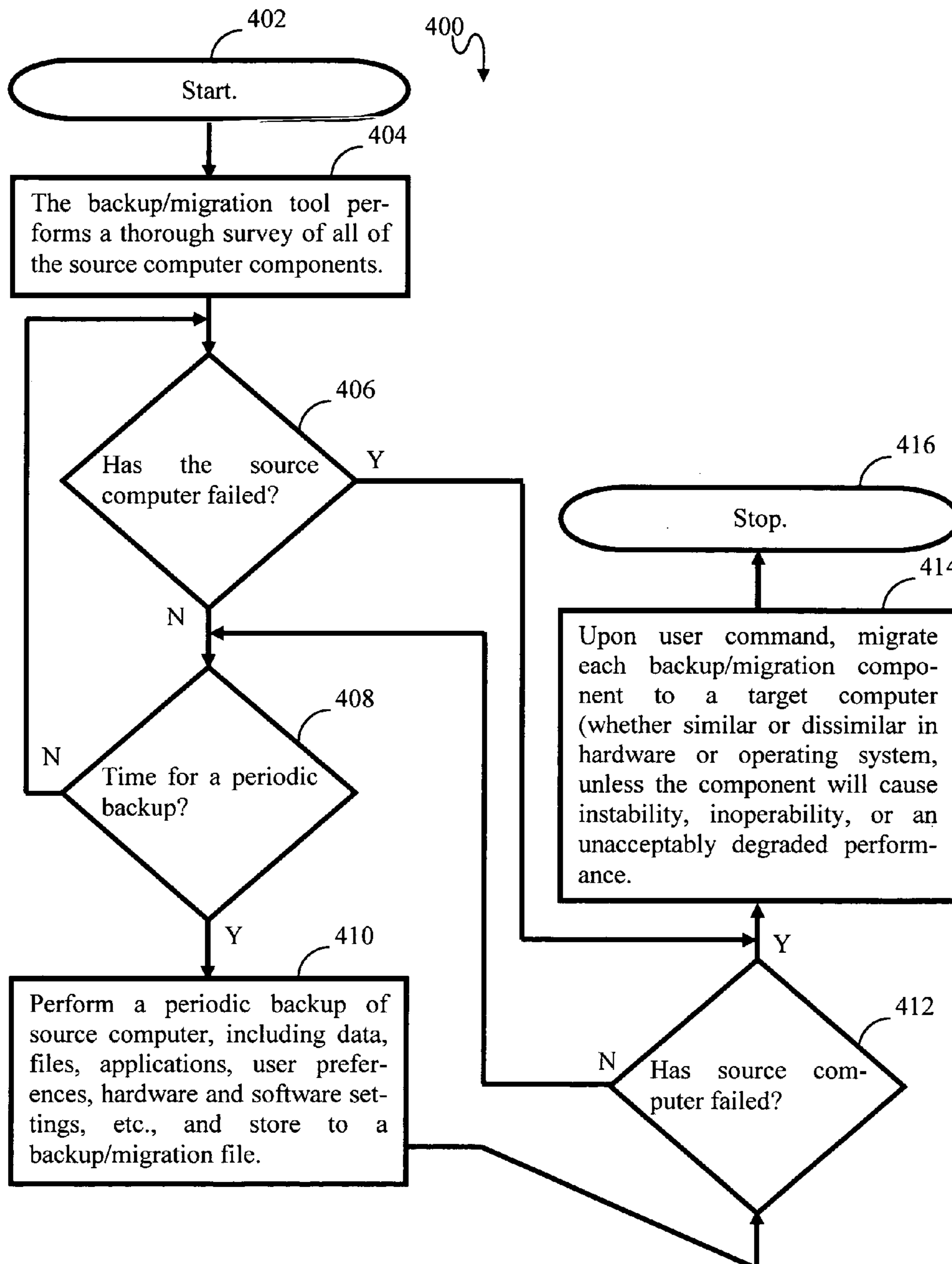


FIGURE 4

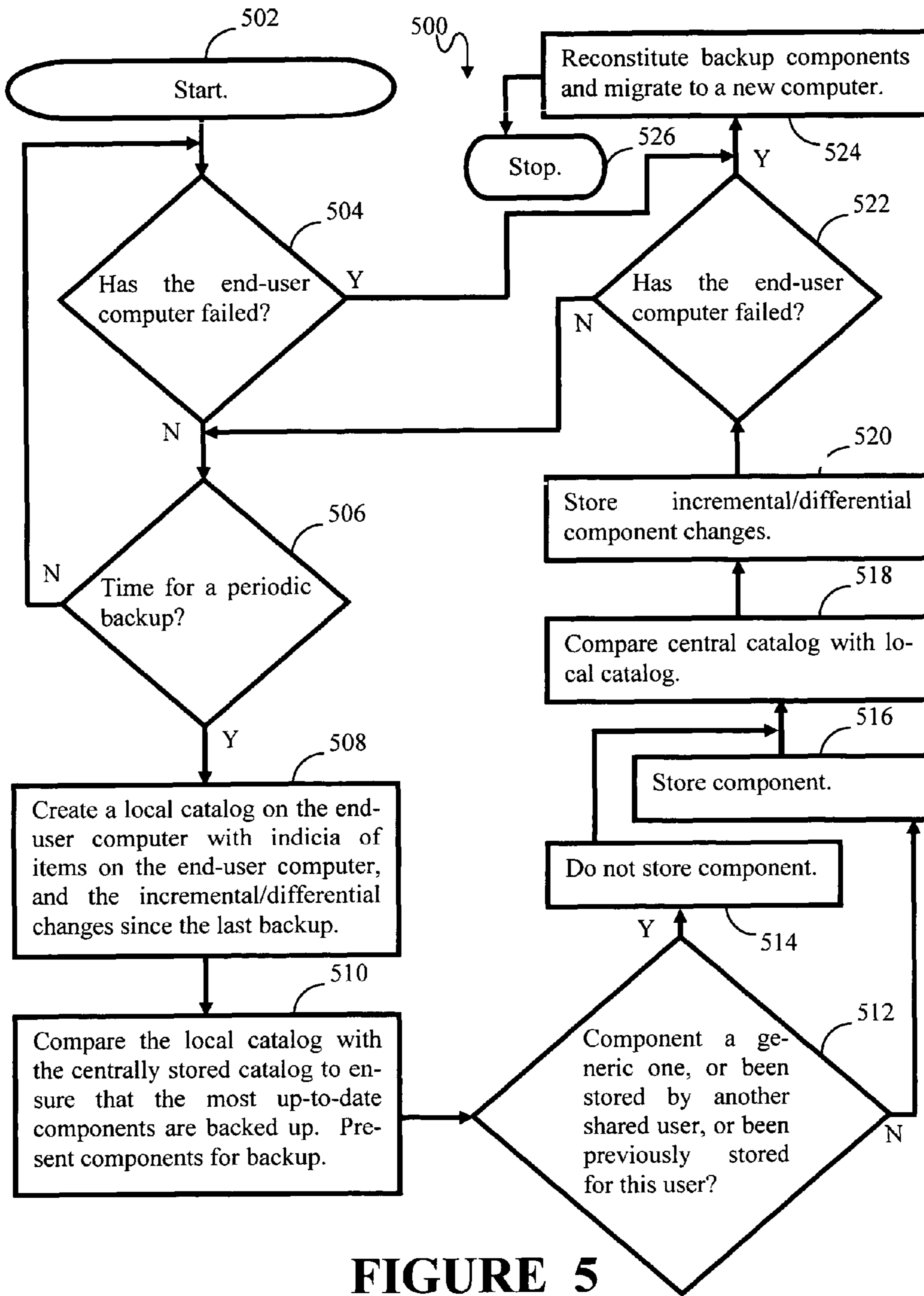


FIGURE 5

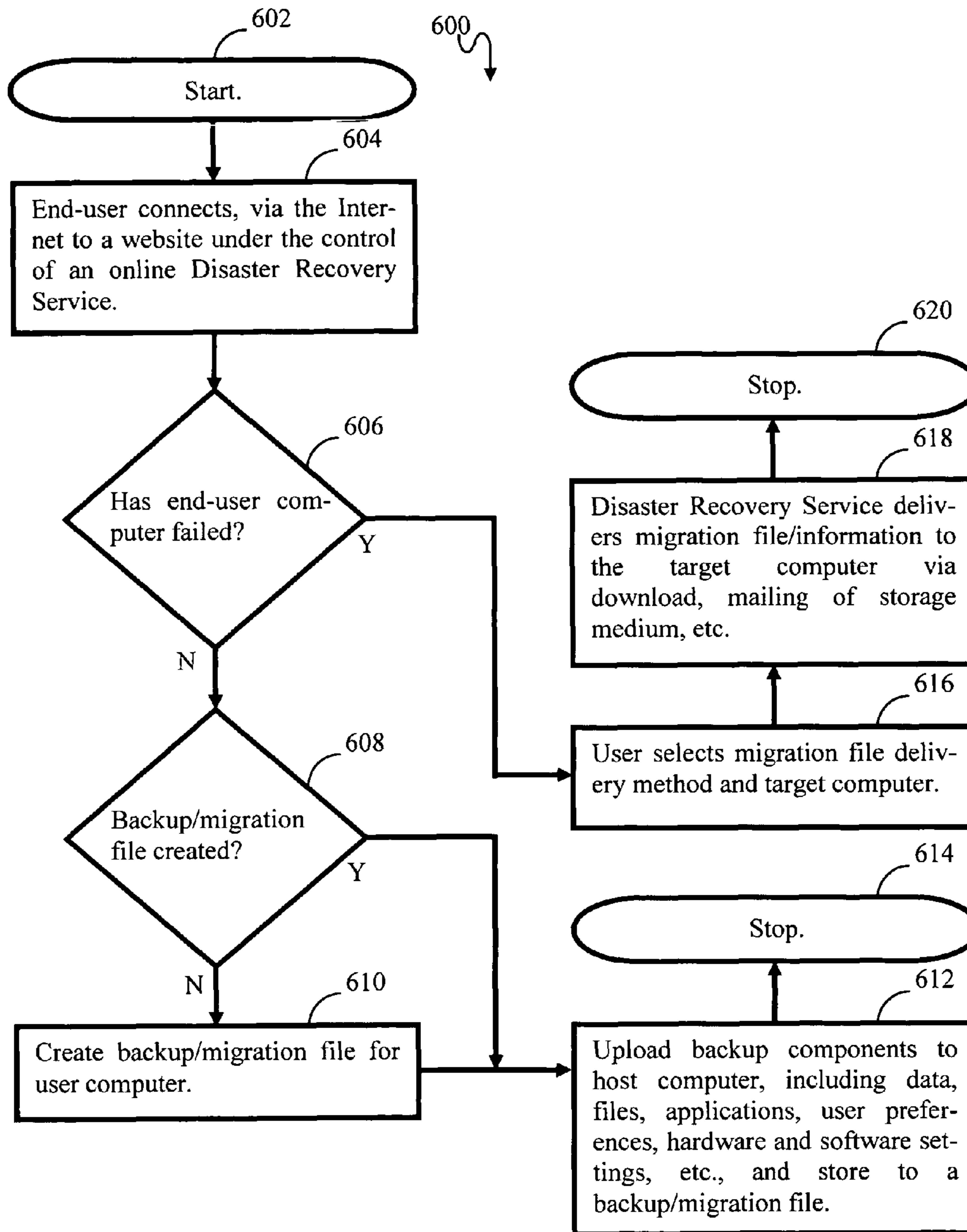


FIGURE 6

1

**COMBINED COMPUTER DISASTER
RECOVERY AND MIGRATION TOOL FOR
EFFECTIVE DISASTER RECOVERY AS
WELL AS THE BACKUP AND MIGRATION
OF USER- AND SYSTEM-SPECIFIC
INFORMATION**

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention broadly relates to computer backup and restore technologies. The present invention also broadly relates to computer migration technologies.

2. Background

The periodic backup of computer system components, such as files, data, applications and the like, is a well-known and oft practiced approach to storing a useful copy of a computer system in a previous state, should the system malfunction in some important way. The storage of the back-up copy can then be used to restore the system to the previous state when the system returns to viability. This may be the most common form of system back-up and restoration.

In the general case, currently available disaster recovery tools create back-up copies of computer systems that can be summoned and transferred to new systems when there is a catastrophic failure. These tools are efficacious for transferring the back-up copy en masse to the new system. However, problems arise when the new system is dissimilar to the old system. When they are sufficiently dissimilar, the transfer can cause problems ranging from system instability, to partial inoperability, to complete inoperability.

Prior art responses to the aforementioned problems include completely overwriting the new operating system with the old one during a full system restore, in order to avoid system conflicts and instability. Using this approach, however, the new system will not be able to take advantage of the new features of the newer operating system, and may not be compatible with newer application software.

To avoid overwriting the new operating system, some prior art approaches merely restore critical data files, leaving out applications, user settings, user preferences, and user state information. Incidentally, none of the prior art disaster recovery approaches is believed to restore user settings, user preferences, and user state information on the new system, whether or not the new system is similar to the old system.

U.S. patent application Ser. No. 09/378,999, filed Aug. 23, 1999, and titled "Application and Method for Transferring Information between Platforms," and which is also assigned to the Assignee of the present Letters Patent, discloses a computer migration method for the efficient transfer of information between a source computer and a target computer. The source and target computers need not have matched hardware or operating system characteristic. Along with data and applications, the aforementioned U.S. Patent Application also allows for the migration of, files, user settings and user preferences, and hardware settings. User-specific settings include internet bookmarks, e-mail settings, names, addresses, telephone numbers, and the like. Hardware-specific settings include the designation of MODEMS, printers, scanners, displays, and the like that are to be used.

The Abstract of the aforementioned U.S. Patent Application states:

A method and apparatus for relocating application programs, settings, menus, files and documents from a source computer to a target computer. The method and apparatus function properly regardless of whether the source and target computers have the same hardware or

2

operating system. The method includes scanning the source and target machines for all applications programs, settings, menus, files, and documents in order to create a relocation strategy. The relocation strategy is created in view of pre-programmed selection rules or selection rules created by the user. The strategy is then implemented by copying, replacing or merging data from the source machine to the target machine.

While the aforementioned U.S. Patent Application discloses useful migration approaches, it does not address disaster recovery, system back-ups used for disaster recovery, catastrophic system failures, and the need to restore the components from a backed-up, failed system to a new system.

Therefore, there is a great unfilled need to provide a tool or tools which can both completely backup computer systems and the like, and to allow the intelligent restoration of the backup information on a target computer system with a dissimilar hardware or operating system, so that the improvements of the target system over the source system are utilized, while maintaining the desirable information and settings from the source system.

The prior art problems associated with disaster recovery and then (as they are not integrated) migration are magnified in a network environment where a server or other central computer must perform these operations for a number of networked end-user computers. Chief among the problems are the astronomical storage space sometimes required to backup each networked computer, and the undue burden on the system resources during the many backup operations that can degrade overall system performance.

The prior art also lacks efficient means of availing safe, robust, and cost-effective backup and migration resources to a large number of computer users who do not have the wherewithal to perform their own extensive periodic backups that can be migrated to a new machine in the case of system failure or upgrade.

SUMMARY OF THE INVENTION

In view of the aforementioned problems and deficiencies of the prior art, the present invention provides an information backup and transfer method. The method at least includes periodically backing up and storing components of a source computer, and in the case of a source computer failure, transferring stored components to a target computer. The method is carried out in an environment where the source computer and target computer are dissimilar. The stored components at least include, as well as critical data files, native information pertaining to the source computer.

The present invention also provides an information backup and transfer system that at least includes a source computer adapted to have its components periodically backed up and stored, a target adapted to have transferred to it, stored components from the source computer in the case of a source computer failure, and a tool for facilitating the backup, storage and transfer of the stored components. The source computer and target computer are dissimilar in either hardware, or operating systems. The stored components at least included, as well as critical data files, native information pertaining to the source computer.

The present invention additionally provides an information backup and transfer method at least including:

- a) networking a plurality of end-user computers to at least one host computer;
- b) via each end-user computer, periodically cataloging the contents of the end-user computer, as well as the state of each

component, and transferring components of the end-user computer to said host computer;

c) via the host computer, periodically backing up components from the end-user computers in a plurality of end-user computer backup profiles;

d) via the host computer, storing and cataloging components which are generic to a plurality of end-user computers;

e) prior to storing a component in an end-user computer profile, noting in the end-user computer profiles, particular generic components that are cataloged on the end-user computers;

f) comparing the state of cataloged end-user components with the state of corresponding generic components;

g) for each end-user computer, storing in the end-user computer profiles those components that are not generic;

h) for each end-user computer, storing in the end-user computer profiles differential backup indicia corresponding to the differences between the host computer version of generic components and the end-user computer version of generic components; and

i) via the host computer, recreating and transferring backup components from a failed end-user computer to a different computer based upon the end-user computer profiles.

The present invention further provides an information backup and transfer system at least including a plurality of end-user computers, and at least one host computer networked to the end-user computers. Each of the end-user computers are adapted to periodically catalog its contents, as well as the state of each component, and transfer components of the end-user computer to the host computer. The host computer is adapted to: periodically back up components from the end-user computers in a plurality of end-user computer backup profiles; store and catalog components which are generic to a plurality of end-user computers; and prior to storing a component in an end-user computer profile, noting in the end-user computer profiles, particular generic components that are cataloged on the end-user computers. The host computer further at least includes a comparator adapted to compare the state of cataloged end-user components with the state of corresponding generic components, and a re-creator adapted to recreate and transfer backup components from a failed end-user computer to a different computer based upon the end-user computer profiles. The host computer is further adapted to, for each end-user computer, store in the end-user computer profiles those components that are not generic, and for each end-user computer, store in the end-user computer profiles differential backup indicia corresponding to the differences between the host computer version of generic components and the end-user computer version of generic components.

The present invention also provides a disaster recovery method that at least includes: a) providing a plurality of remote, end-user computers; b) providing at least one host computer adapted to centrally store end-user computer backup components; and c) providing a wide area networked communication link between the end-user computers and the host computer. The method also at least includes: d) uploading backup information to the host computer; e) via the end-user computers, periodically updating the backup information; and f) in the case of a failure of an end-user computer, transferring backup information pertaining to the failed end-user computer, to a computer designated by a principal of the failed end-user computer.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

Features and advantages of the present invention will become apparent to those skilled in the art from the description below, with reference to the following drawing figures, in which:

FIG. 1 is schematic diagram of a first embodiment of the present-inventive combined disaster recovery and migration system;

FIG. 2 is schematic diagram of a second embodiment of the present-inventive combined disaster recovery and migration system;

FIG. 3 is schematic diagram of a third embodiment of the present-inventive combined disaster recovery and migration system;

FIG. 4 is a flow chart illustrating the first embodiment of the present-inventive method for combined disaster recovery and migration;

FIG. 5 is a flow chart illustrating the second embodiment of the present-inventive method for combined disaster recovery and migration; and

FIG. 6 is a flow chart illustrating the third embodiment of the present-inventive method for combined disaster recovery and migration.

BRIEF DESCRIPTION OF THE APPENDIX

The appendix contains an overview of the processes used for universal recovery according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

General Computer Backup, Disaster Recovery and Migration

A first embodiment of the present-inventive system **100** for disaster recovery and migration is shown in FIG. 1. The system **100** shows a source computer **110** which is the subject of periodic backups, and in the case of catastrophic failure (or a lesser degree of failure determined by the user), will have its essence migrated to a "new" target computer **170**. As will be understood by those skilled in the art, the computers shown nominally include many basic components, including central processing units (CPUs) **120**, **180**. In the environment of the present invention, the source computer and target computer may be dissimilar in their hardware or operating system makeup.

A novel backup/migration tool is shown in FIG. 1 as item **130**. The backup/migration tool **130** contains as its main components, a backup element **140** for conducting periodic backups of the source computer **110**, and a migration element **150** for migrating backed up components to the target computer **170** in the case of a system (**110**) failure. The backup/migration tool **130** is shown in FIG. 1 as a separate external module. However, those skilled in the art to which the present invention pertains will understand that the backup/migration tool **130** can also reside in the source computer **110**. In the latter case, the module **130** could be coupled to an external storage medium (not shown) which could store not only the backup components, but migration software needed during any disaster recovery and migration operation.

Migration software **190** can also be loaded onto the target computer **170** to assist in any migration operation.

The backup and migration files created by the backup/migration tool **130** can be stored externally as part of a capsule to be used by the backup/migration tool **130** to restore the operating system, data, files, applications, user settings, etc. during a disaster recovery and migration operation. The aforementioned components collectively comprise a "user state." That is, the user state is analogous to a comprehensive "snapshot" of a system which reveals the details needed to construct the current status of a system. A novel feature of the present invention is therefore the storage of the user state in backup files, which can be utilized to reconstruct the system

5

on new hardware during a migration operation. The user state is not a mirror image of the entire contents of the drive or other memory components of the system, but selective information that when properly utilized will allow the recreation of the system.

The present invention also novelly treats the user state not as a homogenous set of information, but a group of dependent yet separate layers of information that can be backed up separately, and at separate intervals, depending on the actual or expected amount of change of each component. For example, the operating system may need to be backed up only one time, as it may not be expected to change after it is installed. Data files, on the other hand may need to be backed up frequently, such as once a week, as those files may be expected to change frequently. Still other components such as user settings usually have mid-range dynamics, and might be expected to have a low frequency of variation, but still need regular backing up (e.g., once a month).

It is therefore part of the present invention to not only backup user states, but to employ a rules-based approach to back up different portions of the user state at different frequencies, depending on the actual or expected dynamics of the component. The ability to back up user state components using different backup frequencies allows periodic backup operations to optimize backup file storage space and other system resources by not having to backup all items during every backup operation, and to reduce the time needed for backup operations.

The backup of the components of the source computer **110** can be automated with regard to which components to backup, and the backup frequency for each component. This can be facilitated through the use of scripts or through the use of other approaches, such as the rules-based approach described in U.S. patent application Ser. No. 10/094,251, titled "Non-Script Based Intelligent Migration Tool Capable Of Migrating Software Selected By A User, Including Software For Which Said Migration Tool Has Had No Previous Knowledge Or Encounters," which is also assigned to the Assignee of the present Letters Patent. The Abstract of the aforementioned Application states in part:

[D]etermining how all of the files of a source computer are to be grouped (i.e., to which programs, files are associated) include examining installation log files created by "installation technology," where they exist. Where installation log files do not exist, the migration tool uses a rules-based approach to group files (into programs) according to common folders, common creation dates, common modification dates, and, examining file allocation tables, group files that are proximate. After assets are grouped into Application Groups, a Confidence Level test is performed to determine a degree of confidence that all of the items belonging to the Application Group have in fact been associated with the group. The user is presented with all of the Application Groups and their associated confidence level via an interactive display. The user then chooses programs, files and other assets to migrate to the target computer.

The present-inventive backup/migration tool **130** can serve as the sole backup program for a source computer. Alternatively, the present-inventive backup/migration tool **130** can interface with existing backup tools to create the aforementioned capsule. Both the traditional backup files and the capsule are then used during the migration process to recreate or reconstitute the essence of the source computer **110** on the target computer **170** when needed.

The process employed by the backup/migration tool **130** is illustrated in the flowchart of FIG. 4.

6

After the start **402** of the program **400**, the components of the source computer are classified during an inventory operation (Step **404**) of the backup/migration tool **130** to place each item of the system into specific categories such as applications, settings, operating system and data. These categories aid in determining the frequency that each item will be backed up, as discussed supra.

In the next step **404** the backup/migration tool **130** determines whether the source computer **110** has failed. In case of a source computer failure, the program jumps to Step **414**, where the migration process is started upon the user's command. System failure can be readily determined using known methods and protocols, including non-responsiveness to periodic externally generated signals. As is explained in Step **412**, the source and target computers need not be similar, and components stored in the backup/migration file/capsule are transferred in the preferred embodiment unless they will cause system instability, inoperability, or unacceptably degraded performance. The program ends in Step **416**.

If, however, the system has not failed, the backup/migration tool **130** determines whether a scheduled backup operation is to occur (Step **408**). If it is not time for a periodic backup, the program returns to Step **406**. If it is time for a periodic backup, the backup/migration tool **130** performs a periodic backup of the source computer in Step **410**. As previously mentioned, the backup components include data, files, applications, user preferences, hardware and software settings, menus, directories, etc., as part of the user state.

In the preferred embodiment, Step **410** also determines the nature of changes periodically made to the source computer to gain knowledge of the most efficient way of backing up the components.

After the periodic backup has been performed, the program again determines whether the source computer has failed (Step **412**). In the case of source computer failure, the program advances to Step **414**, described supra. Otherwise, the program returns to Step **408**.

The present invention is also applicable in so-called "mid-disaster" recovery situations. That is, those situations in which a computer is functioning at less than full capacity due to decay or instability in the operating system. In contrast to the prior art approach of perhaps requiring a technician to spend valuable time and resources attempting to diagnose the problem, a master image of a stable system along with any required applications can be applied to the system (thus, also eliminating the unstable operating system), and then re-imaging the backed up user state to restore the system to the previous albeit stable state. The aforementioned approach not only reduces technician time and end-user down time resulting from unstable systems, but is especially valuable in corporate environments, where relatively large numbers of computers may regularly become unstable. Shared Environment Backups Disaster Recovery and Migration

The present-inventive disaster recovery and migration combinations can also be employed in a shared environment, such as in the system **200** shown in FIG. 2. A number of end-user computers (**240**, **260**) can be connected via a local area network (LAN) **230**, or the like to a server **210** which serves to periodically store backup components for each end-user computer. Further, in the case of the failure of an end-user computer, the host/server computer **210** migrates the components of the failed end-user computer to a migration target computer **270**. The system **200** might be used, for example, in a corporate IT/computing environment, where many users are connected via LAN or Intranet to one or more central computers.

As described in the Background Section, *supra*, among the problems associated with prior art shared computing environments with centralized backup (aside from the fact that migrations is not a part of the same tool) is the enormous amount of memory required to backup all of the end-user computers, and the taxing of system resources (and the concomitant system performance degradation) when time-consuming backups occur. In response, the system **200** eliminates or greatly reduces unnecessary backup redundancies, both for each individual end-user computer, and for the system as a whole. This is essentially accomplished by storing for each end-user computer backup/migration file, only what is necessary to reconstitute or recreate the components of a failed system on a new target computer.

The backup/migration files therefore typically contain components only once, and store incremental or differential changes that have occurred since the previous backup of a particular component. This obviates the need for not only storing an entire component twice, but also obviates the need for an often time-consuming overwrite of an existing component, by storing much smaller incremental/differential change information.

Another aspect of redundancy elimination is using linking information in each end-user computer backup/migration file to place a link in the file (rather than the actual item) for all components that already exist in storage elsewhere in the system, whether the component has been stored for another end-user computer, or is a generic component stored and generally used by all or many of the shared users.

Several approaches can be used to determine whether a particular component should be stored in its entirety, or whether incremental/differential information (indicating how the component has changed since it was last stored or considered for storage) should be stored. Data and other components can be classified according to predefined criteria as either static (tending not to change over time) or dynamic (tending to change frequently over time), with static components not needing to be overwritten, and dynamic components needing to be periodically overwritten in between the storage of incremental/differential changes. This is because large changes to a component may make reconstruction of the current status of the component using the original component and incremental/differential information inefficient, and possibly less than fully reliable.

A more flexible approach stores a component for the first time (provided it has not already been stored elsewhere), and thereafter stores incremental/differential information unless certain preconditions exist. One precondition is the passage of a predetermined amount of time. Another precondition is the change of a predetermined amount of the component since it was last stored.

Each end-user computer periodically creates a catalog of all of the backup components as well as the current state of the end-user computer, including the state of each component. When the end-user computer performs a backup operation by transferring components to the host/server, it determines by comparing its catalog to the catalog of stored items on the system **200**, which version of a component is the most up to date. If the local version is more up to date, that version is backed up. If the host/server version of the component is more up to date, the host/server is directed to store that version (or a link to that version) in the end-user computer's backup/migration file.

The host/server **210** in the system **200** nominally contains a unit **220** for backing up and storing components of the end-user computers, a recreation/reconstitution unit for rec-

reating the components of a failed end-user computer, and a migration unit for migrating the recreated components to target computer.

The flowchart in FIG. **5** details the program **500** used by the shared environment aspect of the present invention.

After the start (Step **502**), the program determines whether an end-user computer has failed (Step **504**). If an end-user computer has failed, the program jumps to Step **524**, where the host computer reconstitutes the backup components of the failed end-user computer and migrates them to a target computer. The program stops in Step **526**. If the end-user computer has not failed, the program advances to Step **506**, where it determines whether it is time for a periodic backup of the end-user computer. The program returns to Step **504** if it is not time, but advances to Step **508** if it is time for a periodic backup.

The aforementioned local catalog is created or updated in Step **508**. Prior to transferring components to the host computer, the local catalog is compared with the host computer catalog of all items stored, to determine which versions of the end-user components are most up to date (Step **510**). If the most up-to-date version resides on the end-user computer, it is transferred to the host for storage; if the host version is most up to date, it is stored in the backup file (or a link is placed in the backup file).

In Step **512**, the program determines whether the component is a redundant one. If it is not redundant, it is stored in Step **516**. If the component is a redundant one, it is not stored or restored (Step **514**). The program compares the central catalog with the local catalog (with respect to the component) in Step **518**, and incremental/differential changes to the components are stored where appropriate (Step **520**).

In Step **522**, the program again determines whether the end-user computer in question has failed. If it has failed, the program executes Steps **524** (described *supra*) and **526**. Otherwise, the program returns to Step **506**. Remote Disaster Recovery Service and Migration Information Delivery

In addition to the novel combinations and features described above, the present invention also includes a novel Disaster Recovery Service for the remote storage of end-user computer components, and for the delivery of migration information in the case of system failure to a new end-user computer. This is shown schematically as the system **300** in FIG. **3**.

The essence of this aspect of the present invention is the remote, centralized backup of end-user computers in an application service provider environment. In the system **300**, a source computer can connect to a wide area network **340** such as the Internet to a website **350** controlled by a Disaster Recovery Service. The website **350** is connected to a central backup and migration computer **360** for the backup and migration (in the case of source computer failure) of numerous end-user computers. Those skilled in the art will appreciate that other components may be required for a viable system, depending on the user's environment, such as a LAN, and an Internet Service Provider (ISP).

The computer customer periodically uploads backup components to the central backup and migration computer **360** via an Internet link. If the backed up computer should fail, the customer can connect to the website via another computer, and direct the Disaster Recovery Service to delivery backup and migration information to a target computer **370** of the customer's choice.

The delivery can be in a variety of ways. For example, the central backup and migration computer **360** can download the backup and migration information to the target computer **370**

via an Internet link for the automatic migration of the failed computer's components to the new computer. The Disaster Recovery Service can also deliver the backup and migration information to the customer via traditional mail, or kiosk, etc. on storage media such as CDs, DVDs, ZIPP® disks, etc., for the customer's installation on a replacement computer.

The system 300 implements the program 600 shown in FIG. 6. After the Start (602), the end-user or customer connects to an online Disaster Recovery Service website 350 in Step 604. In Step 606, the program 600 determines (via the interactive information furnished by the customer) whether the customer's backed up computer has failed. If the customer's computer has failed, the program 600 jumps to Step 616, where the customer selects his/her method of delivery of the backup/migration information.

In Step 618, the Disaster Recovery Service delivers the backup/migration information to the new computer or customer (if the customer will install the information), followed by the end of the program (Step 620).

Returning to Step 606, if the customer's computer has not failed, the program 600 determines whether a backup/migration file has been created (Step 608). If the backup/migration file exists, the program 600 jumps to Step 612. Otherwise, a backup/migration file is created in Step 610.

The customer uploads backup components to the central backup and migration computer 360 in Step 612. The program stops in Step 614. Among the components that can be both stored by Disaster Recovery Service and migrated to a replacement computer in the case of a disaster are data, files, applications, user preferences, and hardware and software settings.

In summary, the present invention is a comprehensive, flexible, and robust computer disaster recovery approach which leverages the intimate knowledge gained of the source computer, the knowledge of the target computer, and the knowledge gained through rules and scripts, to allow for efficient, and yet extensive and "personalized" backups, as well as efficient migration to the target computer of the important and personal components of the source computer, while intelligently leaving behind (or allowing the user to choose to leave behind) those elements that will hamper the operation or stability of the target computer, or intelligently leaving behind (or allowing the user to choose to leave behind) duplicate elements which are more contemporary or improved on the target computer.

Stated another way, in contrast to the typical prior art disaster recovery approaches to either: 1) backing up most components of the source computer and transferring the backed up components en masse to a target computer when the source computer fails, which does not allow the user to take full advantage of improvements that may be present in a more recent software that may occupy the target computer, or may cause instability or inoperability; or 2) backing up only "critical" files and data, and not being able to migrate applications, drivers and other important software to prevent potential conflicts between the old system and the new system, neither of which approach stores the "user state" described supra; the present-inventive approach provides greater granularity and flexibility in the backup/disaster recovery/migration processes in a manner that allows the flexible and intelligent selection of the components to backed up to provide as much of a duplicate of the source computer to the target computer as is feasible or desired in the case of disaster recovery, the flexible and intelligent selection of the frequency at which individual components are to be backed up to optimize system resources, and the flexible and intelligent selection of the components to be migrated to the target

computer to provide a new system that contains a personalized legacy of the old system, but with optimized functionality of the new system as desired.

Variations and modifications of the present invention are possible, given the above description. However, all variations and modifications which are obvious to those skilled in the art to which the present invention pertains are considered to be within the scope of the protection granted by this Letters Patent.

What is claimed is:

1. A method, comprising:

determining a nature of changes made to respective components of a user state stored on a source computer;

classifying respective components into at least one of a plurality of backup classes based at least in part upon the determined nature of changes to respective components of the user state;

periodically storing the respective components of the user state on a backup computer according to predetermined schedules associated with the respective backup classes, the respective schedules based at least in part upon at least one of a time that has lapsed since a component was last stored and a percentage change of the component since the component was last stored;

monitoring the source computer to identify at least one of a system failure and a predetermined drop in capacity; and

if at least one of the system failure and the predetermined drop in capacity is identified, transferring at least a portion of the stored user state of the source computer to a target computer that is dissimilar in at least one of hardware and operating system relative to the source computer, transferring at least a portion of the stored user state comprising:

transferring a portion of the user state that will not at least one of hinder an operation of the target computer and cause instability in the target computer from the backup computer to the target computer, the transferred portion of the user state comprising at least one of an application and a user-specific setting, the target computer configured to execute the transferred portion of the user state; and

not transferring a portion of the user state that will at least one of hinder an operation of the target computer and cause instability in the target computer from the backup computer to the target computer,

at least a portion of the method implemented at least in part via a processor.

2. The method of claim 1, the source computer and the target computer comprising end-user devices and the backup computer comprising a device that is shared amongst at least the source computer and the target computer.

3. The method of claim 1, the transferred portion of the stored user state comprising at least one operating system.

4. The method of claim 1, comprising:

identifying a portion of the user state of the source computer that is comprised in a portion of a user state of a second source computer that is stored in the backup computer; and

linking the source computer with the portion of the user state of the second source computer that is stored in the backup computer such that redundant portions of user state from the source computer and the second source computer are not stored in the backup computer.

5. The method of claim 1, comprising, before transferring the at least a portion of the stored user state to the target

11

computer, receiving through a web interface an instruction to transfer at least a portion of the user state to the target computer.

6. The method of claim 1, comprising:

adjusting the predetermined schedules associated with one or more of the backup classes according to knowledge gained regarding a more efficient way of periodically storing the user state on the backup computer.

7. The method of claim 1, comprising:

performing a full backup of the user state to the backup computer during an initial backup; and

subsequent to the initial backup:

performing an incremental backup of at least one component of the user state if a total amount of change since a last backup is less than a predetermined amount, and

performing a full backup of at least one component of the user state if the total amount of change since a last backup is equal to or greater than a predetermined amount.

8. A system, comprising:

a backup component configured to:

determine a nature of changes made to respective components of a user state stored on a source computer; classify respective components into at least one of a plurality of backup classes based at least in part upon the determined nature of changes to respective components of the user state; and

periodically store the respective components of the user state on a backup computer according to predetermined schedules associated with the respective backup classes, the respective schedules based at least in part upon at least one of a time that has lapsed since a component was last stored and a percentage change of the component since the component was last stored; and

a migration component configured to:

monitor the source computer to identify at least one of a system failure and a predetermined drop in capacity; and

if at least one of the system failure and the predetermined drop in capacity is identified, transfer at least a portion of the stored user state of the source computer to a target computer that is dissimilar in at least one of hardware and operating system relative to the source computer, transferring at least a portion of the stored user state comprising:

transferring a portion of the user state that will not at least one of hinder an operation of the target computer and cause instability in the target computer from the backup computer to the target computer, the transferred portion of the user state comprising at least one of an application and a user-specific setting, the target computer configured to execute the transferred portion of the user state; and

not transferring a portion of the user state that will at least one of hinder an operation of the target computer and cause instability in the target computer from the backup computer to the target computer,

at least some of at least one of the backup component and the migration component implemented at least in part via a processor.

9. The system of claim 8, the source computer and the target computer comprising end-user devices and the backup computer comprising a device that is shared amongst at least the source computer and the target computer.

12

10. The system of claim 8, the transferred portion of the stored user state comprising at least one application.

11. The system of claim 8, the transferred portion of the stored user state comprising at least one operating system.

12. The system of claim 8, the backup component configured to:

identify a portion of the user state of the source computer that is comprised in a portion of a user state of a second source computer that is stored in the backup computer; and

link the source computer with the portion of the user state of the second source computer that is stored in the backup computer such that redundant portions of user state from the source computer and the second source computer are not stored in the backup computer.

13. The system of claim 8, comprising a transfer component configured to receive through a web interface an instruction to transfer at least a portion of the user state to the target computer before transferring the at least a portion of the stored user state to the target computer.

14. A computer readable medium comprising instructions that when executed by a processor on a computer perform method, comprising:

determining a nature of changes made to respective components of a user state stored on a source computer; classifying respective components into at least one of a plurality of backup classes based at least in part upon the determined nature of changes to respective components of the user state;

periodically storing the respective components of the user state on a backup computer according to predetermined schedules associated with the respective backup classes, the respective schedules based at least in part upon at least one of a time that has lapsed since a component was last stored and a percentage change of the component since the component was last stored;

monitoring the source computer to identify at least one of a system failure and a predetermined drop in capacity; and

if at least one of the system failure and the predetermined drop in capacity is identified, transferring at least a portion of the stored user state of the source computer to a target computer that is dissimilar in at least one of hardware and operating system relative to the source computer, transferring at least a portion of the stored user state comprising:

transferring a portion of the user state that will not at least one of hinder an operation of the target computer and cause instability in the target computer from the backup computer to the target computer, the transferred portion of the user state comprising at least one of an application and a user-specific setting, the target computer configured to execute the transferred portion of the user state; and

not transferring a portion of the user state that will at least one of hinder an operation of the target computer and cause instability in the target computer from the backup computer to the target computer.

15. The computer readable medium of claim 14, the source computer and the target computer comprising end-user devices and the backup computer comprising a device that is shared amongst at least the source computer and the target computer.

16. The computer readable medium of claim 14, the transferred portion of the stored user state comprising at least one operating system.

13

17. The computer readable medium of claim 14, the method comprising:

identifying a portion of the user state of the source computer that is comprised in a portion of a user state of a second source computer that is stored in the backup computer; and

linking the source computer with the portion of the user state of the second source computer that is stored in the backup computer such that redundant portions of user state from the source computer and the second source computer are not stored in the backup computer.

18. The computer readable medium of claim 14, the method comprising receiving through a web interface an instruction to transfer at least a portion of the user state to the target computer before transferring the at least a portion of the stored user state to the target computer.

19. The computer readable medium of claim 14, the method comprising:

14

performing a full backup of the user state to the backup computer during an initial backup; and

subsequent to the initial backup:

performing an incremental backup of at least one component of the user state if a total amount of change since a last backup is less than a predetermined amount, and

performing a full backup of at least one component of the user state if the total amount of change since a last backup is equal to or greater than a predetermined amount.

20. The computer readable medium of claim 14, the method comprising adjusting the predetermined schedules associated with one or more of the backup classes according to knowledge gained regarding a more efficient way of periodically storing the user state on the backup computer.

* * * * *